

**UNITED STATES DISTRICT COURT  
WESTERN DISTRICT OF TEXAS  
MIDLAND-ODESSA DIVISION**

Malikie Innovations Ltd. and  
Key Patent Innovations Ltd.,

Plaintiffs,

v.

MARA Holdings, Inc. (f/k/a Marathon Digital  
Holdings, Inc.)

Defendant.

Case No. 7:25-cv-00222

**DEFENDANT'S RENEWED MOTION TO DISMISS UNDER RULE 12(B)(6)**

**TABLE OF CONTENTS**

I. Introduction..... 1

II. Background..... 2

    A. The '286 Patent and Montgomery Reductions ..... 2

    B. Malikie’s Infringement Allegations..... 5

III. Legal Standards..... 7

IV. Argument ..... 7

    A. The Reduction Value Limitation Is Material to Practicing the  
        Asserted Claim..... 8

    B. The First Amended Complaint Fails to Plead Infringement of the  
        Reduction Value Limitation..... 9

V. Conclusion ..... 13

**TABLE OF AUTHORITIES**

**Cases**

*Ashcroft v. Iqbal*,  
556 U.S. 662 (2009)..... 1, 7

*Bot M8 LLC v. Sony Corp. of Am.*,  
4 F.4th 1342 (Fed. Cir. 2021) ..... 7

*Disc Disease Sols. Inc. v. VGH Sols., Inc.*,  
888 F.3d 1256 (Fed. Cir. 2018)..... 7

*Qwikcash, LLC v. Blackhawk Network Holdings, Inc.*,  
No. 4:19-CV-876-SDJ, 2020 WL 6781566 (E.D. Tex. Nov. 17, 2020)..... 12

*Vervain, LLC v. Micron Tech., Inc.*,  
No. 6:21-CV-00487-ADA, 2022 WL 23469 (W.D. Tex. Jan. 3, 2022) ..... 1, 7, 8, 9

## I. INTRODUCTION

The Sixth Claim in Plaintiffs Malikie Innovations Ltd.’s and Key Patent Innovations Ltd.’s (“Malikie’s”) First Amended Complaint (FAC)—alleging infringement of U.S. Patent No. 8,532,286—should be dismissed for failure to state a claim. *See Ashcroft v. Iqbal*, 556 U.S. 662 (2009). Malikie’s threadbare allegations of infringement fail to meet the pleading standard for the same reasons this Court articulated in *Vervain, LLC v. Micron Tech., Inc.*, No. 6:21-CV-00487-ADA, 2022 WL 23469, at \*5 (W.D. Tex. Jan. 3, 2022) (Albright, J.).

To state a claim for relief, Malikie’s complaint must allege facts that plausibly suggest that Defendant MARA Holdings, Inc. (“MARA”) has practiced all the claim limitations of at least one asserted claim. All claims of the ’286 patent require “computing a modified operand using a reduction value . . . to perform a replacement of a least significant word of the operand” (hereinafter the “Reduction Value Limitation”). Even taking all of Malikie’s factual allegations as true for the purposes of the pleading stage, Malikie has failed to articulate why it is plausible that MARA practices the Reduction Value Limitation.

Malikie accuses Bitcoin Core’s<sup>1</sup> source code of infringing claim 1 of the ’286 patent. Although the FAC quotes Bitcoin Core’s source code, nowhere does the quoted source code state or suggest that the alleged “reduction value” is used to “perform a replacement of a least significant word of the operand.” Nor do Malikie’s narrative allegations plead a logical connection between the source code and this limitation. In fact, rather than support Malikie’s infringement theory, the quoted source code shows replacement of the *most* significant words of the operand, which is the *opposite* of what is claimed in the Reduction Value Limitation.

---

<sup>1</sup> Bitcoin Core is open-source software for Bitcoin.

The FAC is Malikie’s second failed attempt to plead infringement of the ’286 patent, despite Malikie’s infringement theory being based entirely on publicly available source code files. There is no excuse for Malikie’s inadequate pleading except that MARA does not infringe. For these reasons, as more fully set forth below, Malikie’s claim for infringement of the ’286 patent should be dismissed with prejudice.

## II. BACKGROUND

### A. The ’286 Patent and Montgomery Reductions

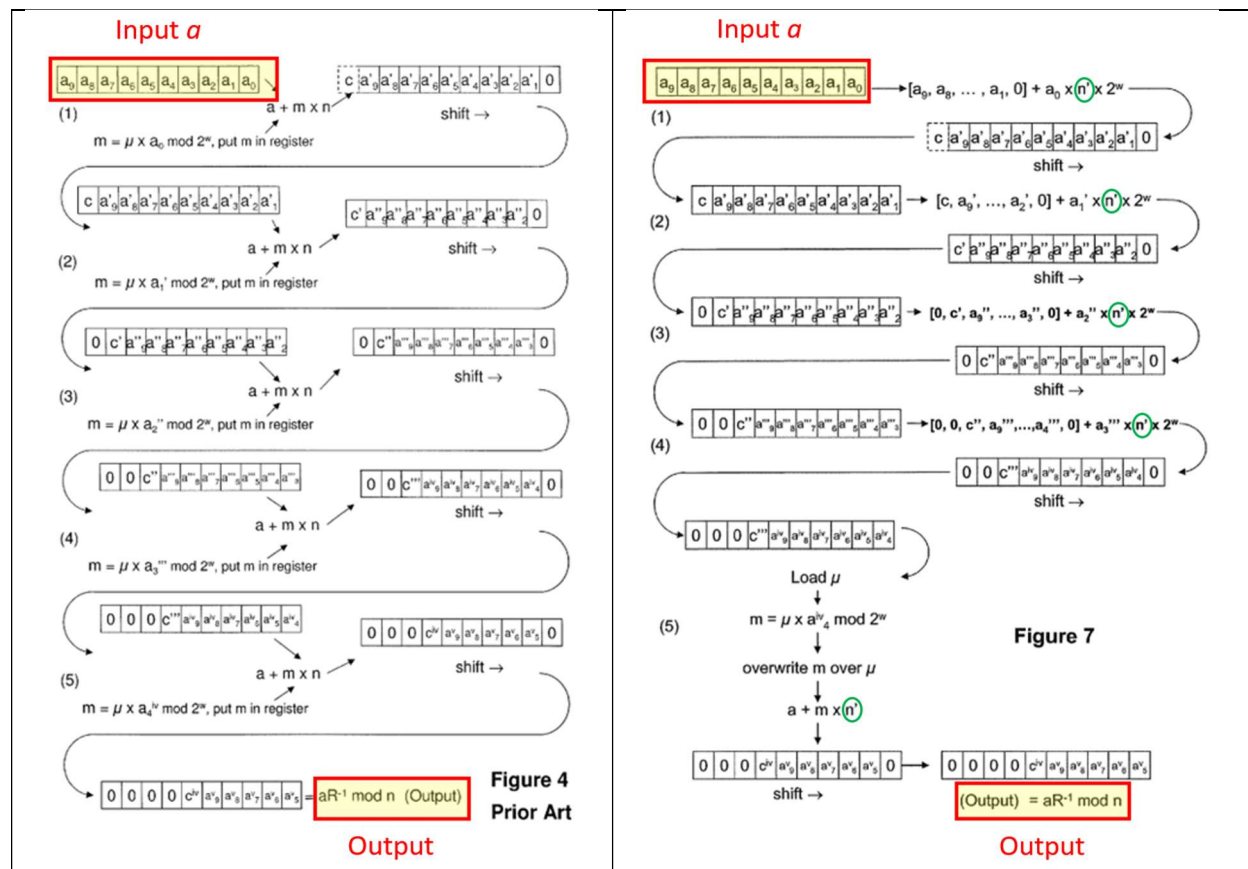
The ’286 patent describes an algorithm for performing a “Montgomery-style reduction.” Montgomery modular reduction, or Montgomery reduction for short, refers to mathematical algorithms for performing modular arithmetic while avoiding computationally expensive division operations. ’286 patent at 1:20–36. Modular arithmetic refers to operations such as addition, subtraction, and multiplication *modulo*<sup>2</sup> some number. “The calculation of the remainder is referred to as reduction in modular arithmetic.” *Id.* Per the ’286 patent, “the step of calculating the remainder is considered slow,” and the “most commonly used” method for modular reduction is Montgomery reduction. *Id.* “Montgomery reduction avoids the expensive division operations typically used in classical modular reduction.” *Id.* The output of a Montgomery reduction as described in the ’286 patent is the value  $aR^{-1} \bmod n$  for a value  $a$  reduced modulo  $n$ .<sup>3</sup> *Id.* at 1:53–59.

---

<sup>2</sup> A modulo or “mod” operation calculates the remainder after one number is divided by another, the latter number being called the “modulus” of the operation. For example,  $5 \bmod 2 = 1$  because a remainder of 1 is left over after dividing 5 by 2.

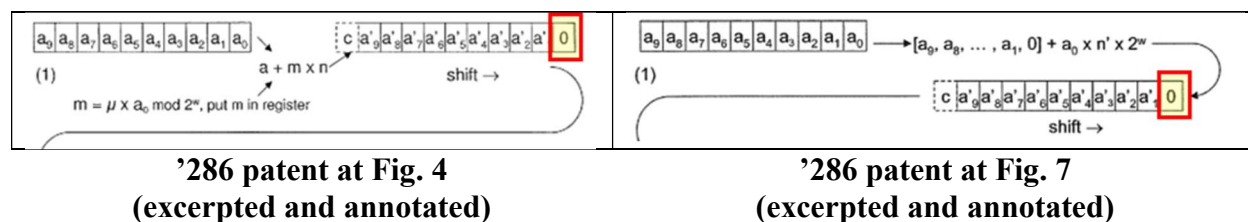
<sup>3</sup> According to the ’286 patent, the value  $R$  is an auxiliary number called the “radix” or “base,” and it can be chosen. ’286 patent at 1:47–52.

The '286 patent describes two algorithms for performing Montgomery reduction: (i) a “typical” or “prior art” algorithm and (ii) a “modified Montgomery reduction using [a] modified reduction value,” which is the purported invention. See '286 patent at 3:1–17, Figs. 3–7. Figure 4 (prior art algorithm) and Figure 7 (modified algorithm), copied below and annotated, illustrate the similarities of the two algorithms.



'286 patent at Fig. 4 (annotated)

'286 patent at Fig. 7 (annotated)



As shown above, in both the prior art algorithm (Fig. 4) and “new” algorithm (Fig. 7):

1. A value  $a$  is the input;
2. The value  $a$  is reduced iteratively (five iterations in the example);

3. At each iteration, the least-significant<sup>4</sup> word<sup>5</sup> of  $a$  becomes “zeroed.” See also ’286 patent at 5:8–10 (“In the result, the least significant word  $a_0$  is ‘zeroed’ . . .”), 6:32–35 (“As shown in FIG. 7 . . . at each iteration, the least significant word of  $a$  is zeroed . . .”); and
4. The output is  $aR^{-1} \bmod n$  (the result of performing a Montgomery reduction on input  $a$ ).

The difference between the admitted prior art algorithm (Fig. 4) and the modified algorithm (Fig. 7) is in the steps performed at each iteration. Specifically, the ’286 patent describes the use of a “modified reduction value”  $n'$  (circled in green in Fig. 7 above). The ’286 patent discloses a single formula for the modified reduction value:  $n' = 2^{-w} \bmod n$ . ’286 patent at 5:45–49. As described in the ’286 patent, the reduction value  $n'$  has the “useful” property that  $1 \equiv n' \times 2^w \bmod n$ , which allows the least significant word ( $a_0$ ) to be “replaced”:

To see the usefulness of this new value, it is noted that if the value  $n'$  is then shifted up by one digit, which is equivalent to multiplying by  $2^w$ , a value is obtained that is equivalent to  $1 \bmod n$ . **Consequently, the value  $a_0$  can be replaced** with  $a_0 n' \times 2^w$ , that is,  $a_0$  multiplied by  $n'$  shifted up one digit. To be explicit  $a \equiv [\dots, a_4, a_3, a_2, a_1, a_0]$  **is replaced with**  $a \equiv [\dots, a_4, a_3, a_2, a_1, 0] + a_0 n' \times 2^w$ .

’286 patent at 5:56–62 (emphasis added). Figure 5 of the ’286 patent also “illustrat[es] the creation of a modified reduction value” (*id.* at 3:10–11):

$$a_0 \equiv a_0[n_4', n_3', n_2', n_1', n_0'] \cdot 2^w$$

’286 patent at Fig. 5 (excerpted)

<sup>4</sup> The “least-significant” figure in a numerical representation may refer to the smallest digit. For example, the “least-significant” digit of the number 5,283 would be “3” (the ones place). The most significant digit would be “5” (the thousands place).

<sup>5</sup> A “word” may refer to a unit of data in a computer. A 32-bit processor, for example, may use 32-bit words. A 160-bit number may be represented as five words on a 32-bit machine (160 divided by 32 is 5). See ’286 patent at 5:40–44.

Claim 1 of the '286 patent, which is the only claim asserted in the FAC, is reproduced below with the Reduction Value Limitation in bold.

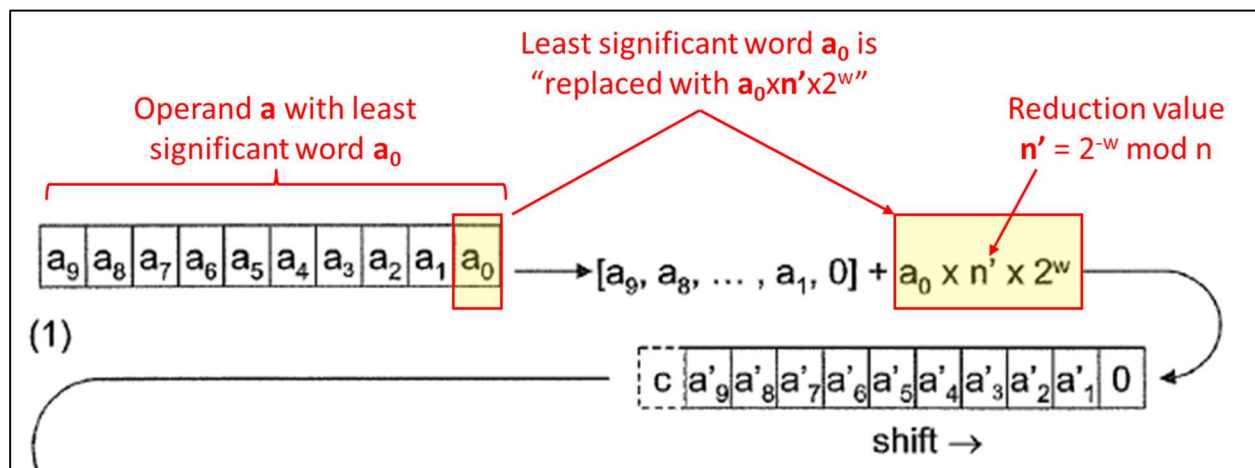
[1Pre] 1. A method for performing, on a cryptographic apparatus, a Montgomery-style reduction in a cryptographic operation, the method comprising:

[1a] obtaining an operand for the cryptographic operation;

**[1b] computing a modified operand using a reduction value, instead of a modulus used in performing a standard Montgomery reduction, to perform a replacement of a least significant word of the operand, rather than perform a cancellation thereof, the reduction value being a function of the modulus; and**

[1c] outputting the modified operand.

In the sole embodiment described in the patent, the “operand” corresponds to a value  $a$  with “least-significant word”  $a_0$ , and the “reduction value” corresponds to  $n'$ , as shown below:



'286 Patent at Fig. 7 (excerpted and annotated)

## B. Malikie's Infringement Allegations

The FAC alleges infringement of claim 1 of the '286 patent under 35 U.S.C. § 271(a) based on MARA's alleged use of “hardware and/or software . . . that comply with the Bitcoin protocol.” FAC ¶ 146. Exhibit 12 to the FAC, which is referenced in ¶ 146, contains an infringement claim chart for claim 1 of the '286 patent.

Malikie's infringement allegations for the Reduction Value Limitation, reproduced below, comprise narrative allegations by Malikie (highlighted in yellow) and three excerpts from the Bitcoin Core source code header file *scalar\_4x64\_impl.h* with some of the code in bold.

|  |   |
|--|---|
| <p>[1b] computing a modified operand using a reduction value, instead of a modulus used in performing a standard Montgomery reduction, to perform a replacement of a least significant word of the operand, rather than perform a cancellation thereof, the reduction value being a function of the modulus; and</p> | <p>MARA computes a modified operand using a reduction value, instead of a modulus used in performing a standard Montgomery reduction, to perform a replacement of a least significant word of the operand, rather than perform a cancellation thereof, the reduction value being a function of the modulus. <i>See, e.g.:</i></p> <p>For example, MARA computes a modified operand (e.g., a value represented in multiple machine words (e.g., "r")) using a reduction value (e.g., using <code>SECP256K1_N_C_0</code>), instead of a modulus used in performing a standard Montgomery reduction, to perform a replacement of a least significant word of the operand (e.g., to perform a replacement of a least significant word (e.g., the value at index "0" of an array of <code>uint64_t</code>) of the operand), rather than perform a cancellation thereof, the reduction value being a function of the modulus (e.g., being a function of a modulus, such as the curve order, e.g., "the secp256k1 order").</p> <pre> /* Limbs of the secp256k1 order. */ #define SECP256K1_N_0 ((uint64_t)0xBF025E8CD0364141ULL) #define SECP256K1_N_1 ((uint64_t)0xBAAEDCE6AF48A03BULL) #define SECP256K1_N_2 ((uint64_t)0xFFFFFFFFFFFFFFFFULL) #define SECP256K1_N_3 ((uint64_t)0xFFFFFFFFFFFFFFFFULL)  /* Limbs of 2^256 minus the <b>secp256k1 order</b>. */ #define <b>SECP256K1_N_C_0</b> (~SECP256K1_N_0 + 1) #define SECP256K1_N_C_1 (~SECP256K1_N_1) #define SECP256K1_N_C_2 (1)  static void <b>secp256k1_scalar_mul</b>(secp256k1_scalar *r, const secp256k1_scalar *a, const secp256k1_scalar *b) {     uint64_t l[8];     ...;     <b>secp256k1_scalar_mul_512</b>(l, a, b);     secp256k1_scalar_reduce_512(r, l);     ...; } </pre> |
|  | <p><i>See, e.g., src/secp256k1/src/scalar_4x64_impl.h (see also code in "scalar_8x32_impl.h")</i></p> <pre> SECP256K1_INLINE static int <b>secp256k1_scalar_reduce_512</b>(secp256k1_scalar *r, const uint64_t *l) {     ...;     secp256k1_uint128 c128;     ...;     uint64_t n0 = l[4], n1 = l[5], n2 = l[6], n3 = l[7];     ...;     /* <b>Reduce</b> 512 bits into 385. */     /* m[0..6] = l[0..3] + n[0..3] * SECP256K1_N_C. */     ...;     /* <b>Reduce</b> 385 bits into 258. */     /* p[0..4] = m[0..3] + m[4..6] * SECP256K1_N_C. */     ...;     /* <b>Reduce</b> 258 bits into 256. */     /* <b>r[0..3]</b> = p[0..3] + p[4] * SECP256K1_N_C. */     secp256k1_u128_from_u64(&amp;c128, p0);     secp256k1_u128_accum_mul(&amp;c128, <b>SECP256K1_N_C_0</b>, p4);     <b>r-&gt;d[0]</b> = secp256k1_u128_to_u64(&amp;c128); ...;     r-&gt;d[1] = secp256k1_u128_to_u64(&amp;c128); ...;     r-&gt;d[2] = secp256k1_u128_to_u64(&amp;c128); ...;     r-&gt;d[3] = secp256k1_u128_to_u64(&amp;c128); ...; } </pre> <p><i>See, e.g., src/secp256k1/src/scalar_4x64_impl.h (see also code in "scalar_8x32_impl.h")</i></p>   |

### III. LEGAL STANDARDS

To state a claim for direct infringement, the plaintiff must plead “factual content” which creates a “reasonable inference” that the accused products meet “each and every element of at least one claim” of the asserted patent. *Disc Disease Sols. Inc. v. VGH Sols., Inc.*, 888 F.3d 1256, 1260 (Fed. Cir. 2018) (quoting *Iqbal*, 556 U.S. at 678).

“The degree of detail required to sufficiently plead direct infringement depends on ‘the complexity of the technology, the materiality of any given element to practicing the asserted claim(s), and the nature of the allegedly infringing device.’” *Micron*, at \*5 (quoting *Bot M8 LLC v. Sony Corp. of Am.*, 4 F.4th 1342, 1353 (Fed. Cir. 2021)). “Under any standard, however, the complaint must support its entitlement to relief with ‘factual content,’ not just conclusory allegations that the accused product(s) meet every claim limitation.” *Id.* at \*2. Providing a claim chart with “element-by-element mapping” is not alone sufficient, rather “[t]he Court must consider whether the factual allegations therein, ‘when taken as true, articulate why it is plausible that the accused product infringes the patent claim.’” *Id.* at \*7; *see also Bot M8*, 4 F.4th at 1354 (“[I]t is the *quality* of the allegations, not the *quantity*, that matters.”).

### IV. ARGUMENT

The FAC utterly fails to articulate why it is plausible that MARA practices the Reduction Value Limitation of claim 1 of the ’286 patent. The level of detail provided in the FAC does not meet the standard demanded of the Reduction Value Limitation, which is material to practicing the asserted claim and in fact contains the only alleged point of novelty over the admitted prior art. Malikie’s claim chart merely parrots the claim language and provides excerpts of public source code files that bear no resemblance to the Reduction Value Limitation. At most, the cited source

code shows the *opposite* of what is claimed, i.e., “replacement of a *most* significant word of the operand.”

The Sixth Claim of the FAC should be dismissed with prejudice because Malikie has, for the second time, failed to plausibly allege infringement of the '286 patent.

**A. The Reduction Value Limitation Is Material to Practicing the Asserted Claim**

The specification and prosecution history confirm that the Reduction Value Limitation is the alleged point of novelty for claim 1 of the '286 patent. Accordingly, the Reduction Value Limitation is “material to practicing the asserted claim.” *Micron*, at \*5.

First, as discussed above in Section II.A, the specification shows that the admitted prior art Montgomery reduction algorithm (Fig. 4) is substantially similar to the purported invention (Fig. 7). The sole difference is in the use of a particular “reduction value.” *See also* FAC ¶ 109 (describing the use of a particular “reduction value” as the “inventive technique”).

Second, the prosecution history confirms that the Reduction Value Limitation “lays at the point of novelty.”<sup>6</sup> The applicant distinguished claim 1 over prior art based on the Reduction Value Limitation. Ex. 1 (October 10, 2012 Applicant Remarks) at 11 (“Sabin may teach details of a Montgomery Reduction, however, Sabin does not teach or fairly suggest the use of a modified reduction value, let alone as recited in claim 1.”); Ex. 2 (May 1, 2013 Applicant Remarks) at 7 (“A modified process cannot be considered equivalent to a reduction value unless the process is modified using such a value.”); *id.* at 7 (“[N]ot only is the claimed method used to perform a reduction differently, a reduction value is used, as clearly recited in claim 1.”); Ex. 3 (May 13, 2013 Notice of Allowance) at 2 (“[T]he prior art fails to teach or reasonably suggest the invention

---

<sup>6</sup> The Court may take judicial notice of the prosecution history of the '286 patent. *Micron*, at \*5 & n.2.

as claimed. For example, see the applicant's remarks filed on 5/1/13, which contrasts the cited references against the invention as claimed.”).

Thus, based on statements made during prosecution, in the FAC, and in the '286 patent itself, the Reduction Value Limitation is material to practicing the asserted claims.

## **B. The First Amended Complaint Fails to Plead Infringement of the Reduction Value Limitation**

The level of detail provided in the FAC does not meet the pleading standard here, “where the technology is not simple and the limitations-at-issue are material.” *Micron*, at \*5.

First, the narrative statements by Malikie in the claim chart attached to the FAC do not explain how the accused products could plausibly infringe the Reduction Value Limitation. As shown in Section II.B above, the claim chart contains two narrative statements for the Reduction Value Limitation [1b]:

MARA computes a modified operand using a reduction value, instead of a modulus used in performing a standard Montgomery reduction, to perform a replacement of a least significant word of the operand, rather than perform a cancellation thereof, the reduction value being a function of the modulus. *See, e.g.:*

For example, MARA computes a modified operand (*e.g.*, a value represented in multiple machine words (*e.g.*, “r”)) using a reduction value (*e.g.*, using SECP256K1\_N\_C\_0), instead of a modulus used in performing a standard Montgomery reduction, to perform a replacement of a least significant word of the operand (*e.g.*, to perform a replacement of a least significant word (*e.g.*, the value at index “0” of an array of uint64\_t) of the operand), rather than perform a cancellation thereof, the reduction value being a function of the modulus (*e.g.*, being a function of a modulus, such as the curve order, *e.g.*, “the secp256k1 order”).

FAC, Ex. 12 at 10. The first statement is conclusory and “merely track[s] the claim language.” *Micron*, at \*7. The second statement identifies accused *variables* in the source code but does not state where or how those variables actually perform the Reduction Value Limitation. For example, the FAC identifies a “reduction value” and a “least significant word of the operand” but fails to

identify where or how the alleged reduction value is used “to perform a replacement of a least significant word of the operand.”

Second, the claim chart includes three block excerpts to public Bitcoin Core source code, with some code in bold, but Malikie does not plead a logical connection between the code and the Reduction Value Limitation [1b]. None of the excerpted code bears any resemblance to the “reduction value” described and claimed in the ’286 patent, and, for all three excerpts, the relevance of the bold code is unclear and unexplained.

Based on the source code quoted in the FAC, Malikie alleges that the “least significant word of the operand” is index “0” of array **l**, or **l[0]**. Specifically, for limitation [1a], Malikie identifies a variable named “**l**” as the alleged “operand”:

For example, MARA obtains an operand, e.g., a value (e.g., “1”) represented in multiple machine words (e.g., 8 machine words). The operand is for the cryptographic operation (*see* [1pre], *supra*). *See, e.g.:*

#### FAC, Ex. 12 at 8

The FAC quotes source code showing a variable called “**l**” being defined as an array of type “uint64\_t” and size “8”:

```
static void secp256k1_scalar_mul(secp256k1_scalar *r, const secp256k1_scalar *a, const
secp256k1_scalar *b) {
    uint64_t l[8];
    ...;
    secp256k1_scalar_mul_512(l, a, b);
    secp256k1_scalar_reduce_512(r, l);
    ...;
}
```

#### FAC, Ex. 12 at 9 (annotated)

Malikie also describes **l** as “represented in multiple machine words (e.g., 8 machine words).” FAC, Ex. 12 at 8. The eight alleged “words” of array **l** are, presumably, **l[0]**, **l[1]**, **l[2]**, **l[3]**, **l[4]**, **l[5]**, **l[6]**, and **l[7]**. Then, for limitation [1b], Malikie identifies “the value at index ‘0’ of an array of

uint64\_t” as the alleged “least significant word of the operand.” FAC, Ex. 12 at 10. Index “0” of array **l** is **l[0]**. Thus, Malikie is alleging that the “least significant word of the operand” is **l[0]**.

In the source code quoted by the FAC, there is just one reference to **l[0]**—the alleged “least significant word of the operand”—which is in a comment highlighted below:

```
SECP256K1_INLINE static int secp256k1_scalar_reduce_512(secp256k1_scalar *r, const
uint64_t *l) {
    ...;
    secp256k1_uint128 c128;
    ...;
    uint64_t n0 = l[4], n1 = l[5], n2 = l[6], n3 = l[7];
    ...;
    /* Reduce 512 bits into 385. */
    /* m[0..6] = l[0..3] + n[0..3] * SECP256K1_N_C. */
```

#### FAC, Ex. 12 at 11 (annotated)

The highlighted comment contains an equation: “ $m[0..6] = l[0..3] + n[0..3] * SECP256K1\_N\_C$ .”

The variables in that equation correspond to the following allegations in the FAC:

- **SECP256K1\_N\_C** is the alleged “reduction value.”
- **l[0..3]** are the four least significant words of the alleged operand **l**.
- **n[0..3]** are the four most significant words of the alleged operand **l**. As shown in the source code excerpt above the highlighted comment, the variables named “**n0**,” “**n1**,” “**n2**,” and “**n3**” are defined as “uint64\_t” variables corresponding to **l[4]**, **l[5]**, **l[6]**, and **l[7]**, respectively, i.e., the four most significant “words” of alleged operand **l**.

Even assuming, in the absence of any explanation, that Malikie is alleging infringement of “perform[ing] a replacement of a least significant word of the operand” based on the above source code comment, such an allegation is contradicted by the comment itself. Unlike the ’286 patent, the comment highlighted above shows the alleged reduction value being multiplied with the *most*

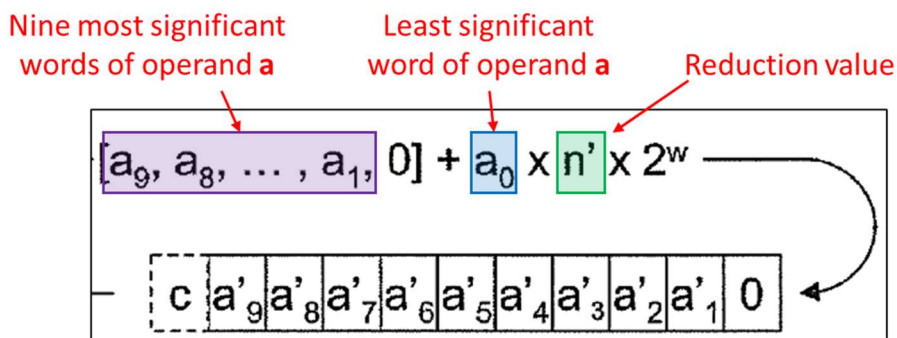
significant words of the alleged operand. Compare the below, annotated excerpt of the accused source code:

```
uint64_t n0 = l[4], n1 = l[5], n2 = l[6], n3 = l[7];
...;
/* Reduce 512 bits into 385. */
/* m[0..6] = l[0..3] + n[0..3] * SECP256K1_N_C. */
```

Four least significant words of alleged operand l
Four most significant words of alleged operand l
Alleged reduction value

**FAC, Ex. 12 at 11 (annotated)**

with Figure 7 of the '286 patent:



**'286 Patent at Fig. 7 (excerpted, annotated)**

Accordingly, the quoted comment shows that the alleged “reduction value” SECP256K1\_N\_C is used to perform a replacement of the *most significant* words of the alleged operand l.

Thus, not only does Malikie fail to adequately allege that MARA practices the Reduction Value Limitation (which renders the FAC deficient), but the documents cited in the FAC demonstrate that Malikie has no basis to allege that MARA practices this limitation. *See Qwikcash, LLC v. Blackhawk Network Holdings, Inc.*, No. 4:19-CV-876-SDJ, 2020 WL 6781566, at \*4–5 (E.D. Tex. Nov. 17, 2020) (granting motion to dismiss plaintiff’s claims of infringement due to the “irreconcilable internal inconsistencies on the face of its pleading” regarding a key limitation,

and “further demonstrated by [plaintiff’s] failure to provide a plausible allegation that [defendant’s] system performs at least the [key limitation]”).

**V. CONCLUSION**

Malikie has failed to plausibly allege infringement of the ’286 patent. Accordingly, MARA respectfully requests that the Court dismiss the Sixth Claim of the FAC with prejudice.

Dated: August 7, 2025

Respectfully Submitted,

/s/ Steve Wingard

Steve Wingard

State Bar No. 00788694

swingard@scottdoug.com

Robert P. Earle

State Bar No. 241245566

rearle@scottdoug.com

Stephen L. Burbank

State Bar No. 24109672

sburbank@scottdoug.com

Scott Douglass & McConnico LLP

303 Colorado Street, Suite 2400

Austin, TX 78701-3234

Telephone: (512) 495-6300

Facsimile: (512) 495-6399

Anish R. Desai

Elizabeth S. Weiswasser

Ian A. Moore

Tom Yu

Thomas Macchio

Paul, Weiss, Rifkind, Wharton & Garrison  
LLP

1285 Sixth Avenue

New York, NY 10019

Telephone: (212) 373-3000

Christopher M. Pepe

W. Sutton Ansley

Eric C. Westerhold

Paul, Weiss, Rifkind, Wharton & Garrison  
LLP

2001 K Street NW

Washington, DC 20006

Telephone: (202) 223-7300

*Counsel for Defendant MARA Holdings, Inc.*

**CERTIFICATE OF SERVICE**

I hereby certify that I have served a true and correct copy of this motion upon each attorney of record and the original upon the Clerk of Court on this the 7th day of August, 2025.

/s/ Steve Wingard  
Steve Wingard